National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Data Structures**	**Course Code:**	**CS 218**
Program:	**BS (CS)**	**Semester:**	**Fall 2023**
Type:	**Assignment 1**	**Section:**	**A**

Important Instructions:

Submit separate .cpp files in ZIP folder. The naming format of each file Should be: 22L-RollNo._Q_No.cpp i.e. 22L-XXX_Q_X.cpp. The files violating the Format won't be considered for grading. Late Submissions won't be accepted.

Question no. 1

Given a singly linked list and a positive integer k, write a function to rearrange the nodes in the list such that the nodes are grouped by k nodes into sub-lists. Within each sub-list, the nodes should be in their original relative order.

Example:

Input:

LIST: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9
k = 3

Output:

[1 -> 2 -> 3] -> [4 -> 5 -> 6] -> [7 -> 8 -> 9]

If the list cannot be divided exactly on k, the last sub-list will be of the remaining nodes that are less than k.

Note: The output should be a linked list of linked lists, where each sub-list is represented as a separate linked list node.

Question no. 2

You are given a file named **input.txt** with values Yes and No. Read the file and insert the strings in a 2D array of **strings**, where the elements are either "Yes" or "No". Do not use extra space. You must calculate number of rows and columns after reading the file. Now, your task is to compress the input array by counting the number of "Yes" values and creating a new array, which will contain the count of "Yes" values in the first column and the string "No" in the second column. Then, you need to use the compressed array to transform it back to the original input array. You can use another 2D array as an intermediate representation. Additionally, you need to calculate the number of "Yes" values in the original array and print the result. Make an input function to make Input 2D array from user and fill it up

Yes	Yes	Yes No No No
Yes	No	No No Yes No

Yes	No	No No No No
Yes	Yes	Yes No No Yes
Yes	No	No No Yes No

Output Array:

3 3

2 4

1 5

4 2

2 4

In this 2D array the first row will contain the location of "Yes". For example, the first row will look like this 1, 2, 3. Which represents that "Yes" in first row are at location 1, 2 and 3 and rests of the elements are "No". **Use string not cstring.**

You need to solve this problem in $O(\log_2 N)^2$

Question no. 3

In a computer system the main memory (RAM) is a shared resource that all running programs share. The operating systems serves as a resource manager. The main job of theoperating system for memory management includes: allocation of free memory to the programs whenever they request it, reclaim the memory of the programs after finishing their execution and keeping record of free and allocated memory.

Programs can claim memory at the start of the program and during its execution. So the memory reserved by one program may not be contiguous. Similarly, due to dynamic nature of the programs, new programs are added to the system and existing programs are removed after finishing their execution. This may cause memory fragmentation. **As processes are loaded and removed from memory, the free memory space is broken into little piecesis called fragmentation.** In the figure below shaded areas are free memory scattered in themain memory.

The job of an operating system is to allocate available byte(s) when a new

process(program) is created and deallocate reserved memory when an existing program isfinished. A program in memory is stored in multiple bytes. The Memory management module of the operating system maintains a **pool** of available bytes (or contiguous chunks of bytes) where new data of an existing program or a new program can be stored. It also store the list of programs currently in execution and the parts of memory allocated to them.Whenever a new program is started, the memory manager uses available bytes of RAM tostore the data of that program on the disk. Similarly when a program is finished, the memory(bytes) allocated to that program are moved back to the pool of available memory.

Whenever a program needs memory, the memory manger asks the total number of bytes required to store the data. The memory manager finds a chunk of contiguous memory in RAM and allocate that memory to the program. If the program asking memory is a new program a new program is added to the list of current programs. Otherwise the information of the block of bytes allocated to the existing program is kept with the program information. A block maintains the Id of starting byte and total number of bytes. So a program in a memory management system has a collection of blocks where each block maintains the address of next block of the program and program stores the address of first block. In order to make the whole process efficient, the memory management system maintains the *pool of available blocks instead of available bytes*.

Your task is to simulate the memory management system. The Memory management system includes: RAM, list of available blocks (pool) and list of Programs currently in execution. We define the RAM as a one-dimensional array of bytes of size numOfBytes. A pool of available blocks is kept in a linked list of block. A block has two fields: start_byte_ID, and total_bytes. Programs are kept in a linked list Progs where each node has a program as data. Each program has an ID, and a linked list of blocks allocated to that program. Asking memory requires claiming a sufficient number of bytes from pool, if available and update the pool accordingly. Removal of a program requires removing the list of blocks allocated to the program and transferring the bytes assigned to this program back to pool.

Suppose we have a RAM of size 100KB. This means we will have byte ID's from 0 to 102400-1.

Figure 1 shows a link list of blocks called pool. The first available block is of 30 bytes starting from byte ID 420 and last block is of 9900 bytes starting at byte ID 4000. Figure 2 shows a sample link list of Programs. There are 3 programs in the memory. The Id of program1 is P1 currently has acquired 260 bytes. It also tells that the data of the program is stored in 3 blocks. First block starts at byte ID 0 and has 40 bytes, second block has 20 bytes that starts from byte ID 70 and last block also has 200 bytes starting from byte ID 120.

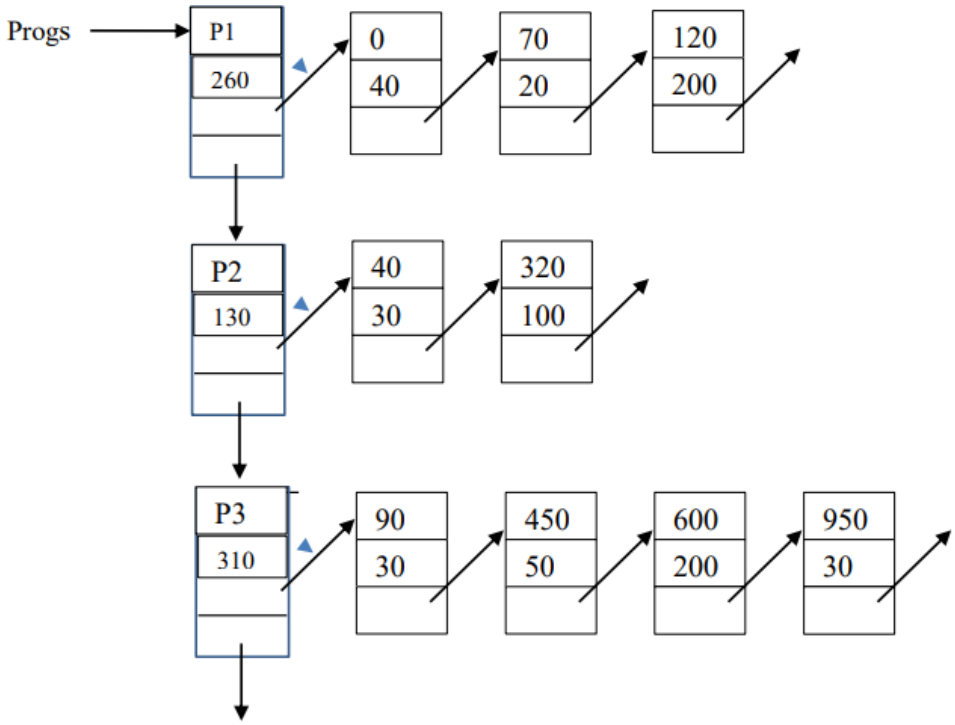


Figure2: Programs list containing three Programs P1, P2 and P3

In order to implement the memory management system you need to implement the structure given in classroom.

Member functions:

GetMem: The GetMem function must take program Id and size of the memory required. as parameters and should work as follows:

Find the memory of required size from the pool of free memory using strategy s (will be explained later). If the requested memory is not available then return false. Otherwise, Id is searched in the list of programs. If the program Id already exists then a new block of memory is removed from the pool of free memory and added to the existing program at theend of the program linked list of blocks. If no such program already exists then first a newprogram is inserted at the end of Progs and then this block is added to the list of blocks ofthe newly inserted program.
The memory management system implements one of the two strategies:

First fit strategy:
The memory management system allocates the first available block of memory that fits the program requirement.

Best fit strategy:
The management system allocates the free block of free memory that has smallest size and meets the requirement of the program.

Efficient fit strategy:
Let say user assigned a new program and we already used memory(bytes) from all blocks of pool, then our getMem function will return false. Then you need to ask user whether you want to allocate memory forcefully or not after checking whether you can allocate memory forcefully or not. If user say yes, then you will take some bytes from one block and some bytes from other block and so on until you fulfill the total size requires by new program. This case will trigger in both strategies (First and Best fit).

For example a new program P4 requests a block of 200 size. According to the first fit strategy, the memory will be allocated starting at 980 byte ID. This block has 400 bytes. The first 200 bytes will be allocated to P4 and remaining 200 resides in the pool. The updated pool and Progs list is shown the figure 3 and figure 4.

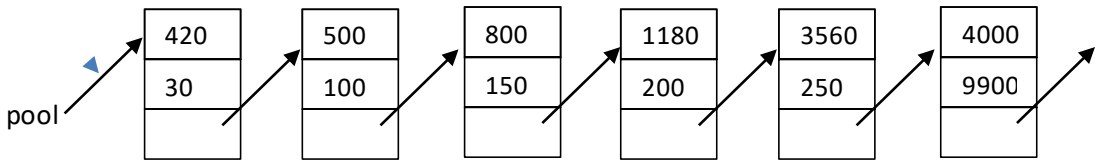


Figure3: Updated pool of blocks using first fit strategy

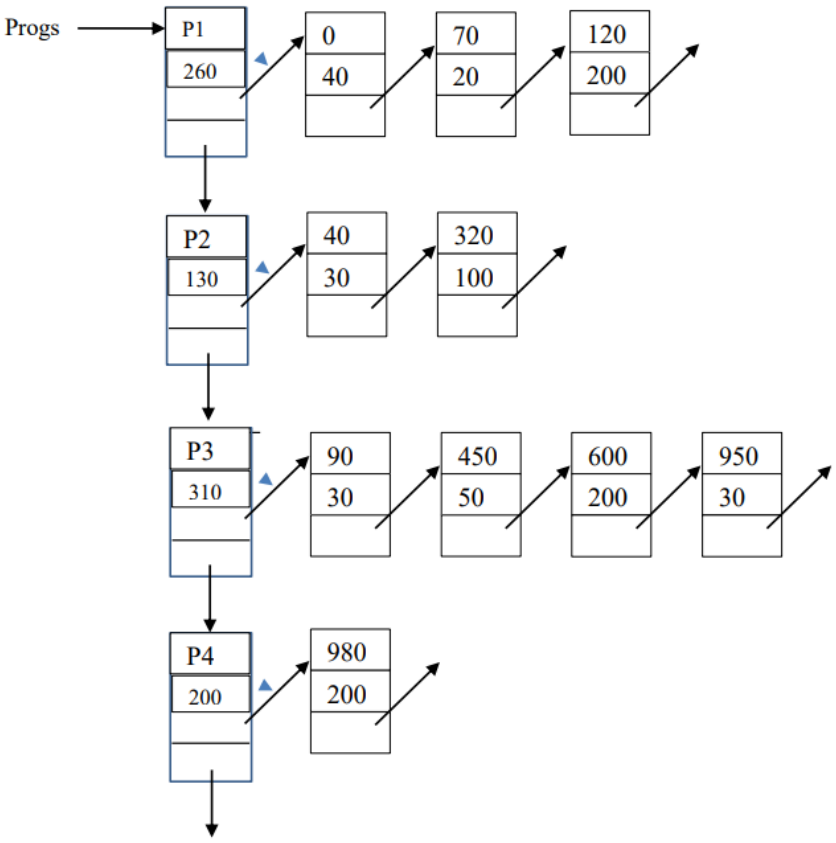


Figure4: Updated Progs list using first fit strategy

However if the strategy is best fit then the block starting at 3560 byte will be allocated.The updated pool and Progs is shown in Figure 5 and 6.

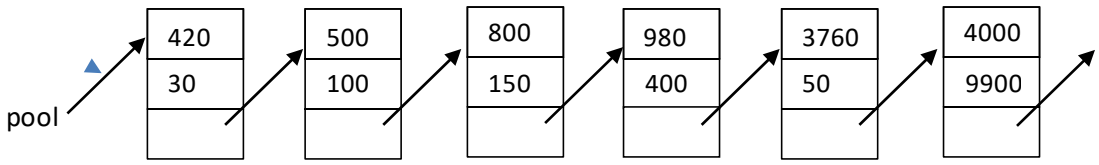


Figure5: Updated pool of blocks using best fit strategy

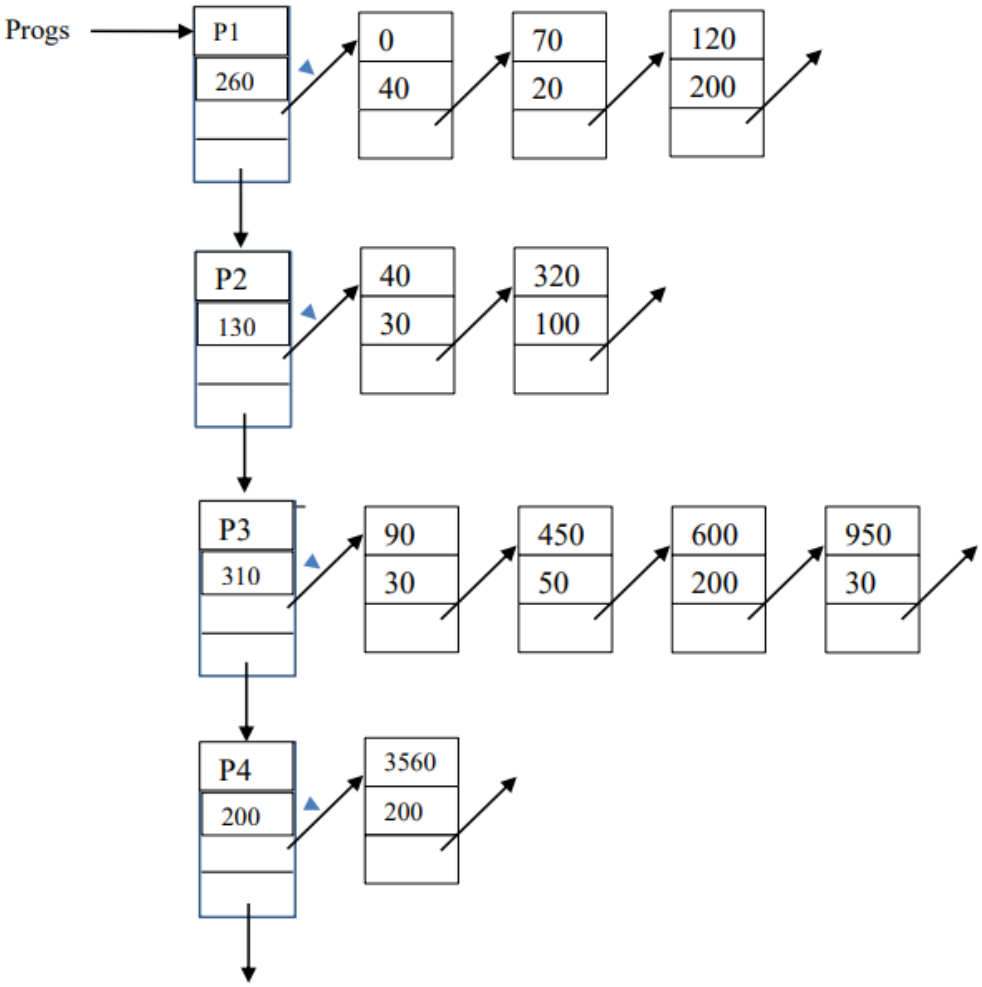


Figure6: Updated Progs list using best fit strategy

DeleteProgram:

This function must take program Id as parameter and delete the program with the given Idfrom Progs. Also move all the blocks of the memory to pool in sorted order. If two consecutive block become one single contiguous block then you must also merge them into one single block. For example if program P3 is deleted then the block starting at 450 must be merged with the block in pool starting at byte position 420 and so on. The updated pool will be as follows

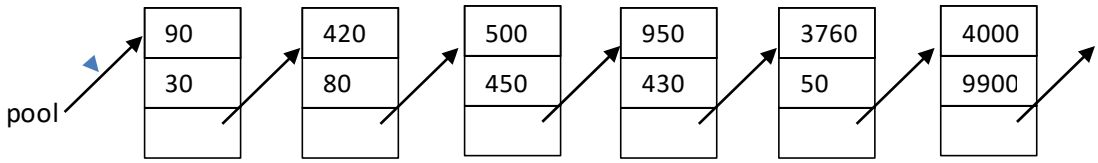


Figure5: Updated pool of blocks after deletion of P3

Constructor:

Initially Progs is empty and pool has only one block. SizeofMemory and Strategy will be initialized to user provided values and will remain the same throughout the program.

Provide a driver function that creates a memory management system object and a menuthat prompts the user to perform different operations of the memory management.

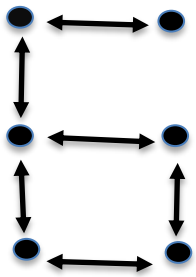
Question no. 4

You all may have noticed the digits displayed on calculators and digital clocks. Every digit is derived from the digit '8'.

Consider that every point where two or more lines meet in a node. The structure of a single digit will be as following where each black dots represents a node.

Each node has 4 pointers (node* left, node* right, node* up, node* down). Any digit can be written using these 6 nodes, by establishing essential links.

For example, 6 can be made as:



The link is two way because it does not matter you link it left to right or right to left.
You will receive a time input from the user in a 24-Hours format.

Your task is to:

Make separate linked list for each digit, and print the time.

Sample Input:

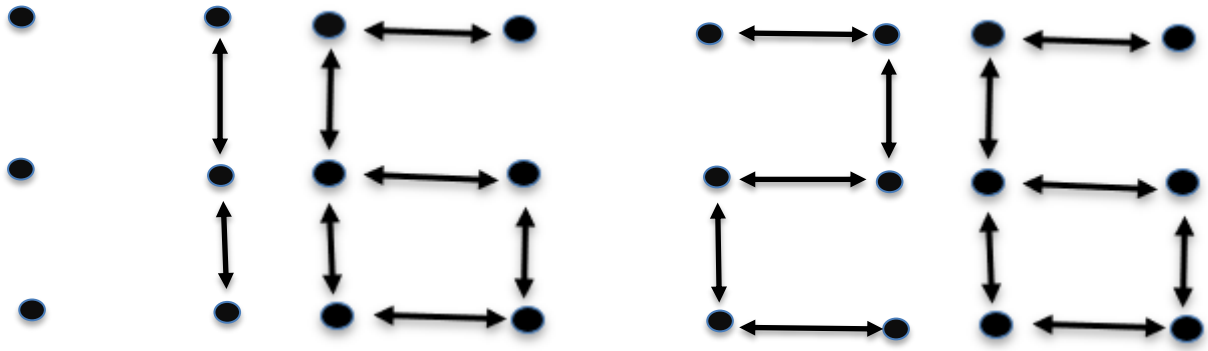
16:26

Output:

```
*  *******     *******  *******
*  *                 *  *
*  *******     *******  *******
```

```
*  *******     *******  *******
*  *     *     *         *     *
*  *******     *******  *******
```

Internal Structure:



You need to make sure that all the pointeres are properly handled. Every pointor that is not pointing to any of the node should be pointing to null.

FUNCTIONAL IMPLEMENTATIONS:

1. Addition of Minutes

The function should take the minutes input from the user, add it in the time and print the time after addition of respective minutes.
NOTE: Addition of minutes may affect hours.

2. Subtraction of Minutes

The function should take the minutes input from the user, subtract it from the time and print the time after subtaction of respective minutes.
NOTE: subtraction of minutes may affect hours.

3. Addition of Hours

The function should take the hours input from the user, add it in the time and print the time after addition of respective hours.

4. Subtraction of Hours

The function should take the hours input from the user, subtract it from the time and print the time after subtaction of respective hours.

5. Conversion to 12 hour Format

The function should display the entered time in 12 hour format along with AM/PM.

6. Change of Date

The program should display a date changed message if any of the functionalities above, when performed on time, results in the changing of date.